**Al Amin Choyon| Full Stack & AI-ML Python Developer**

[choyonwork@gmail.com](mailto:choyonwork@gmail.com) **|**

**PROFESSIONAL SUMMARY**

Experienced Software Developer with around 6 years of experience in software development. Passionate about backend development and optimization. Strengths include Spring framework migration, and full-stack proficiency in Python, C++, PostgreSQL, Node, React, Java, JavaScript and Spring Framework.

* **IT experience** in developing web-based applications, software development and design using Python 3.3/2.7, Django 1.9/1.8, XML, CSS, HTML, DHTML, JavaScript, jQuery & Angular.
* Experienced in MVC frameworks like Django, Angular JS, jQuery, React.js and Node.js. Flask and Pyramid, Well versed in Django ORM and SqlAlchemy and Extensive use of Linux (Ubuntu), SSH, Flask/Django.
* Proficient in SQL databases MS SQL, MySQL, and no-SQL databases MongoDB, Cassandra
* Worked on AJAX framework to transform Datasets and Data tables into HTTP-serializable JSON strings.
* Experience in developing Pyspark scripts from source system like Azure Event Hub to ingest data in reload, append, and merge mode into Delta tables in Databricks.
* Built scalable microservices using Flask/Django with MVC architecture, designed REST APIs for React/JavaScript frontends, and containerized applications with Docker for AWS deployment.
* Expertise in creating Restful API in NodeJS and communicate with Clojure server via protocol and use Backbone to generate template.
* Developed predictive models with scikit-learn, performed data analysis using Pandas/NumPy, and visualized insights via Matplotlib in Jupyter notebooks
* Experience in performing the load testing using JMeter.
* Have Extensive experience in learning and practices newer tools like Solr, Elastic search, Kibana, Lucerne, Spotfire. Built Analytics KPI engine using Python and PIG.
* Deployed containerized apps on AWS (EC2, Lambda, S3), automated CI/CD pipelines, and monitored performance using CloudWatch.
* Experienced in developing web-based applications using Python, Django, XML, CSS, HTML and JavaScript.
* Performed complex data analysis and provided critical reports to support various departments.
* Familiar with Web Development, Amazon Web Services, Python and the Django framework
* Very strong experience writing API's/ and Web Services in PHP and in Python and utilized Python Pandas Framework to provide data analysis.
* Leveraged VSCode for full-stack debugging, optimized API performance, and implemented automated testing for robust microservices.
* Expertise in writing Jenkins file for creating dynamic pipeline to support Continuous Integration and Continuous Delivery.
* Well versed in working with Version Control Systems/Source Code Repository tools like Subversion, GIT, CVS, SVN GitHub, Performance maintaining/troubleshooting the CM tool in UNIX and Windows environments.
* Strong understanding of Agile and Scrum Software Development Life Cycle Methodologies.
* Experienced with setup, configuration and maintain ELK stack (Elasticsearch, Logstash and Kibana) and OpenGrok source code (SCM)

**SKILLS CHART:**

**Languages:**Python 3.x,2.7/2.4, C++, Java, Shell Script, Perl, SQL, JavaScript

**Python Framework:**Django (1.3/1.4/1.5), Flask, FastAPI, Pyramid

**Databases: Relational:** MySQL, PostgreSQL, Oracle, SQLite, **NoSQL:** MongoDB, Cassandra, Redis, Elasticsearch

**Web Technologies:**HTML5, CSS3, JavaScript (ES6+), AJAX, jQuery, React.js, AngularJS, Vue.js , REST API, GraphQL, Microservices, MVC, WebSockets , XML, JSON, YAML

**AI/ML Technologies:**Pandas, NumPy, SciPy, Scikit-learn, Matplotlib, Seaborn, Jupyter, TensorFlow (Basic), Keras (Basic)

**Cloud & DevOps : AWS** EC2, S3, Lambda, RDS, ECS, CloudWatch, IAM, SNS, SQS, CI/CD  Jenkins, GitHub Actions, AWS CodePipeline , Docker, Kubernetes, Docker Compose, Terraform (Basic), AWS CloudFormation

**Versioning Tools:**Git, GitHub, GitLab, Bitbucket, SVN (Subversion), CVS, Perforce

**IDE:**VS Code, PyCharm, Eclipse, MyEclipse, RAD, Jupyter Notebook, Spyder

**Web servers:**Apache HTTP Server, Nginx, IIS, Gunicorn, uWSGI

**Testing & Debugging tools:**Selenium, PyTest, UnitTest, Postman, Swagger, Insomnia, ELK Stack (Elasticsearch, Logstash, Kibana), Prometheus, Grafana,

**Message Brokers & Streaming** : Kafka, RabbitMQ, AWS SQS, Redis Pub/Sub

**Tools/Platforms:**PySpark, Apache Airflow (Basic) , OAuth2, JWT, SSL/TLS,  Bash, PowerShell, AWS, Docker

**Professional Experience**

**Spectrum, Stamford, CT Nov 2023 to Present**

**Role: Python Developer**

**Project:** Subscriber Activity Insight

**Spectrum** is the trade name of Charter communications. Services that Spectrum offers include cable television, internet access, and internet security, managed services, mobile phone, and unified communications. This project was developed as part of strategic initiative to identify peak usage hours and optimize network resources accordingly. Also to detect early signs of customer churn and suggest retention strategies.

* Extract, transform, and load (ETL) telecom data from various sources (network logs, customer usage records, billing systems).
* Used **Pandas, NumPy**, and **SQL** to clean and structure data for analysis.
* Optimized data pipelines for efficiency and scalability.
* Built APIs and microservices using **Flask** to serve data to the dashboard.
* Implemented authentication and access control for telecom data.
* Developed analytical models to track **network performance, customer behavior, and revenue trends**.
* Developed **Flask microservices** with **REST API** endpoints for telecom data processing, deployed on **AWS** using **Docker** containers for scalability.
* Built **predictive models** using **scikit-learn** and **Pandas** to analyze subscriber churn patterns, visualizing trends with **Matplotlib/Seaborn** dashboards.
* Optimized **PostgreSQL** queries and implemented caching to accelerate real-time analytics for network usage insights.
* Automated ETL pipelines with **Python** to ingest/transform telecom data from network logs into structured **NumPy/Pandas** datasets.
* Designed secure **MVC** architecture with **JWT authentication** for subscriber activity dashboards using **React/JavaScript**.
* Used **Matplotlib, Seaborn, and Plotly** to create interactive visualizations based on the requirements for dashboard.
* Optimized queries and data retrieval for real-time analytics.
* Implemented caching mechanisms to reduce load times.
* Document code, workflows, and data structures for maintainability.
* Developed entire frontend and backend modules using Python on Django Web Framework.
* Worked on designing, coding, and developing the application in Python using Django MVC.
* Developed and executed test plans, Test scenarios, Test cases, Test Scripts for Manual and Automation Testing.
* Performed troubleshooting, fixed, and deployed many Python bug fixes of the two main applications that were a main source of data for both customers and internal customer service team.
* Tested API integrity to ensure proper functionality using Postman.
* Worked with POSTMAN for API testing. Developed REST APIs, and created User Model for application.
* Developed Docker container based micro services using Spring Boot and deployment on Kubernetes cluster.

**Environment:** Python, Django, HTML, CSS, XML, AWS, Elasticsearch, PostgreSQL, GraphQL, Pandas, NumPy, Oauth, **Matplotlib, Seaborn, and Plotly**

**Citibank, Dallas, TX Feb 2022 to Oct 2023**

**Role: Python Developer**

**Project:** Loan Analysis & Risk Assessment Modernization

Primary objective of this project was to modernize the loan application by analyzing loan applications and assess risk using **Python-based data science and machine learning techniques.** By leveraging historical loan data, Citibank could make informed decisions, minimize defaults, and optimize lending strategies. This system was used to evaluate loan applications in real time. If a borrower has a high probability of default, the system flagged the application for further review, reducing financial risk.

* Participated in requirement gathering and analysis phase of the project in documenting the business requirements by conducting workshops/meetings with various business users.
* Extracted and cleaned loan application data using **Pandas** and **NumPy**.
* **Used Exploratory Data Analysis (EDA) to** Identify trends in borrower profiles, income levels, and default rates.
* **To implement the Risk Prediction Models,** used **Random Forest, Logistic Regression, and SVM** to predict loan default probabilities.
* Optimized model accuracy by selecting key financial indicators.
* Used **Matplotlib and Seaborn** to generate insights for financial analysts.
* Implement form handling in **React**, including the management of form state with hooks (useState) and managing submission via **APIs**, often integrating with **Django backend** APIs.
* **Developed risk assessment microservices** using **Flask** and **REST API** principles, containerized with **Docker** and deployed on **AWS EC2** for scalable loan analysis.
* **Built machine learning models** with **scikit-learn** (Random Forest/Logistic Regression) to predict loan defaults, leveraging **Pandas/NumPy** for feature engineering and data preprocessing.
* **Created dynamic dashboards** with **Matplotlib/Seaborn** visualizations to present risk insights, integrated with **React/JavaScript** frontend via secure API endpoints.
* **Optimized PostgreSQL** schemas and queries for high-performance financial data analysis.
* **Automated data pipelines** using **Python** to extract/transform loan application data from multiple sources into structured formats for ML modeling.
* Leverage **AWS** services to scale and deploy web applications, including **EC2** instances for application hosting, **S3** for static file storage (images, documents), and **RDS** for scalable database hosting.
* Use **AWS Lambda** to build serverless functions for background tasks or event-driven architecture, such as processing file uploads or interacting with **AWS SNS/SQS** for messaging.
* Set up and manage CI/CD pipelines using AWS CodePipeline, CodeDeploy, or third-party tools (e.g., **GitHub** Actions, **Jenkins**) to automate deployment to Elastic Beanstalk or ECS.
* Implement authentication and authorization in Django using built-in Django authentication mechanisms or third-party libraries like **JWT (JSON Web Tokens)** or OAuth2, while ensuring secure API access for React frontend via tokens.
* Optimize backend performance with Django, including using Query Optimization, Caching (e.g., **Redis**, **Memcached**), and Database Indexing to improve the response time of API calls.
* Monitor and maintain application health using **AWS CloudWatch** for logging, metrics, and setting up alarms for issues like resource over-utilization or API errors.
* Collaborate with cross-functional teams to integrate UI/UX designs into React components, ensuring that the application delivers a smooth and user-friendly experience.

**Environment:** Python, Pandas, NumPy, Scikit-learn, Matplotlib, Seaborn, Django, HTML, CSS, XML, AWS, Elasticsearch, PostgreSQL, GraphQL,

**Wells Fargo, NYC May 2021 to Jan 2022**

**Role: Python Developer**

**Project:** Automated Financial Data Extraction & Analysis

Goal of this project was to develop a **web-based financial data extraction and analysis system** to streamline risk assessment, investment insights, and regulatory compliance for Wells Fargo. By automating data retrieval and processing, the platform enhances decision-making efficiency and reduces operational costs. Business reasons and benefits included identification of potential risks in lending and investment portfolios, process automation which provides operation efficiency, Regulatory Compliance by maintaining accurate records and Audit ready reports and investment insight.

* Develop and maintain backend services using **Django** and **Django REST Framework**, including **APIs**, data models, and business logic.
* Design and implement **RESTful APIs** to support frontend applications and external integrations.
* Build and maintain **frontend components** using **React**, ensuring responsive and dynamic user experiences.
* Integrate **frontend** and **backend** systems, handling API consumption and ensuring smooth data flow between client and server.
* Write unit and integration tests for both frontend (**React Testing** Library/Jest) and backend (**Django/DRF** testing tools) to ensure application reliability.
* Work with databases (e.g., **MySQL**, **PostgreSQL**), including schema design, migrations, and query optimization.
* **I**mplement authentication and authorization mechanisms, often using tools like Django’s auth system, JWT, or OAuth2.
* Leverage **AWS services** for deployment and scaling, including **EC2** for hosting, **RDS** for database management, **S3** for static file storage, and Lambda for serverless functions.
* Monitor and manage application performance on **AWS** using tools like **CloudWatch** for logging, metrics, and alerts.
* Deploy and manage application infrastructure using AWS services such as **Elastic Beanstalk**, **ECS**, or **EKS** for container orchestration.
* Collaborate with cross-functional teams including designers, **QA**, and **DevOps** to deliver high-quality software.
* Stay updated with modern development trends, libraries, and **frameworks** in the **Django**, **React**, and **AWS** ecosystems.

**Novartis, NC October 2019 to April 2021**

**Role: Python Developer**

**Project:** Clinical Trial Data Management System

Objective of the project was to develop a web-based platform to streamline the management, analysis, and visualization of drug discovery data, enabling researchers to make data-driven decisions efficiently. Key features of the project were Collect and process experimental results from various sources, Provide real-time visualization of compound performance, Enable researchers to share insights and findings securely & Generate detailed reports for regulatory compliance.

* Created the backbone of the application by building **Django Views** for data processing and response handling. Define and interact with data through **Django Models**, ensuring your database structure is both efficient and scalable.
* Build secure, scalable **REST APIs** using **Django REST Framework**. Utilize **serializers** for transforming data between the backend and frontend, and implement **viewsets** to simplify CRUD operations.
* Develop interactive, user-friendly interfaces with **React**, breaking down the UI into functional components that use **React Hooks** like useState and useEffect to manage internal state and lifecycle logic.
* Use **useState** for local state management in React and leverage **Redux** or **React Context** for handling global state across multiple components, ensuring a consistent user experience.
* Use **Django Templates** to render dynamic content on the server-side when required, while using React for client-side interactivity, offering a seamless combination of static and dynamic rendering.
* Secure your APIs and frontend interactions by implementing **authentication** in Django using **JWT tokens**. Ensure the React frontend securely manages user sessions and makes authenticated API requests.
* Deploy your application using **AWS EC2** instances for hosting, **S3** for storing static assets, and **RDS** for managed database services. Scale your application horizontally to handle high traffic loads.
* Integrate **AWS Lambda** for executing serverless functions, allowing you to perform tasks like image processing or background jobs without the need for dedicated server resources.
* Streamline your development process with automated build and deployment pipelines using **AWS CodePipeline**. Deploy your application seamlessly to **Elastic Beanstalk** or **ECS**, ensuring high availability and low downtime.
* Fine-tune Django API performance by leveraging **caching**, **query optimization**, and **indexing**. In React, improve performance by using **React.memo** and **code-splitting** to optimize component rendering.
* Use **AWS CloudWatch** to monitor application health and troubleshoot performance bottlenecks. Set up alarms for key metrics to catch potential issues before they affect users.
* Work closely with DevOps and front-end teams to ensure smooth integration across the stack. Follow best practices for writing maintainable code and participate in regular **code reviews** to ensure high-quality output.

**Environment:** Python, Bash, REST, JSON, Django, MYSQL, Apache, Amazon EC2, Lambda function, React JS.

**EDUCATION:**

CUNY York College, Jamaica, NY

**Major:** Computer Science, B.S (GPA: 3.79)

**Programming Coursework**: Data Structures, Principles of Programming Languages, Operating Systems, Software Development, Network Computing

**CS Coursework**: Automata and Computability, Assembly Language Programming, Computer Architecture